Index

ShareWare Tools

V1.0

ShareWare Tools is a library of functions ShareWare authors can use to prevent people from using their programs beyond the trial period, and to simplify the registration process. It includes 9 functions, a sample program, and two tools to make the functions easier to use.

Functions

These are the functions that make up the SWTOOLS library, along with their parameters and, index number, and a brief description. The way they are declared here is how they would be declared in Turbo Pascal for Windows 1.5; for other programming languages , consult the manual for information on using DLLs. To find out more about a specific function, click on its name.

function Scramble (What, Password, Result :PChar; v :byte) :integer; far; external 'SWTOOLS' index 1;

Scrambles *What* with *PassWord* and returns the result in *Result* . It returns a 1 if successful.

function CreateDateFile (FileName, Password :PChar; v :byte) :integer; far; external 'SWTOOLS' index 2;

Creates a file called *FileName* containing the current date, scrambled with *Password* . It returns 1 if successful.

function CreateNewDate (FileName, Password :PChar; v :byte) :integer; far; external 'SWTOOLS' index 3;

Creates an empty date file called *FileName* . When DaysGoneBy reads an empty date file, it writes over it with one containing the current date. Returns 1 if successful.

function DaysGoneBy (FileName, password :PChar; v :byte) :integer; far; external 'SWTOOLS' index 4;

Returns the number of days that have elapsed between the date in *FileName* and the current date.

function CheckRegistration (FileName, Name, Password :PChar; more :integer; var MoreToFollow :PDosStream; v :byte) :integer; far; external 'SWTOOLS' index 5;

Checks the registration file called *FileName* with the password *Password* and returns 1 if it is valid, 0 if it is not. To read more information from the registration file, set *more* to nonzero. In this case, it will return a pointer to a DOS stream in *MoreToFollow* , which you should pass to GetRegInfo and DoneWithInfo.

function CreateRegistration (FileName, Name, Password :PChar; more :integer; var MoreToFollow :PDosStream; v:byte) :integer; far; external 'SWTOOLS' index 6;

Scrambles *Name* with *Password* and places this information in a file called *FileName* . To include more information, pass *More* as nonzero and pass *MoreToFollow* to WriteRegInfo and DoneWithInfo. Returns the same number that was passed in *More* .

function GetRegInfo (MoreFollowing :PDosStream; var ExtraInfo :PChar; v :byte) :integer; far; external 'SWTOOLS' index 7;

Gets an additional piece of information from the registration file passed in *MoreFollowing* and puts it in *ExtraInfo* . Returns the length of *ExtraInfo* .

function WriteRegInfo (MoreFollowing :PDosStream; ExtraInfo :PChar; v :byte):integer; far; external 'SWTOOLS' index 8;

Writes the string passed in *ExtraInfo* to the stream pointed at by MoreFollowing. Returns the length of *ExtraInfo* .

function Scramble (MoreFollowing :PDosStream; v:byte):integer; far; external 'SWTOOLS' index 9;

Closes the file pointed at by *MoreFollowing* and frees all associated memory. Returns 1 if successful.

See also: Date Stamping and Registration tools .

Scramble function

```
function Scramble (What, Password, Result :PChar; v :byte) :integer; far; external
'SWTOOLS' index 1;
```

Description:

This is the function used by all of the functions that require a password to scramble information or to check the validity of it. This prevents users from modifying their registration and date stamp files, to "counterfeit" a registration or set the date stamp back. It is not necessary to call this function yourself to use the other functions, but you may find it useful. You can also use the Scramble tool to access this function.

Parameters:

What : Whatever is to be scrambled, up to ten characters. With the date stamping functions, it is the date; with the registration functions, it is the name of the person registering. Scramble ignores the case of this parameter.

Password : The word used to scramble it, up to 8 characters. This should be a word or series of characters unique to each of you programs. Scramble does NOT ignore the case of the Password.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : 1 if the function is completed.

CreateDateFile function

```
function CreateDateFile (FileName, Password :PChar; v :byte) :integer; far; external
'SWTOOLS' index 2;
```

Description:

This function is called by DaysGoneBy when it reads an empty date file . It creates a date stamp file containing the date on which it was executed. Your program does not need to call this function unless it wants to handle its own date stamping.

Parameters:

FileName : The name and path of the date stamp file to be created. CreateDateFile will NOT add any extension to this file.

Password : The date is scrambled with this password and written, along with it, to the file. This is to prevent someone from simply setting the numbers in their date stamp file back whenever their time runs out.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns :1 if the function is completed.

See also: Date stamping .

CreateNewDate function

```
function CreateNewDate (FileName, Password :PChar; v :byte) :integer; far; external
'SWTOOLS' index 3;
```

Description:

This function creates an empty date file . When such a file is read by DaysGoneBy, it calls CreateDateFile, which stamps it with the current date. Files created by CreateNewDate are the original date stamps that should be distributed with your program. You do not have to call this function in order to use date stamping; it can be carried out by the MakeDate tool.

Parameters:

FileName :The name and path of the new date stamp file to be created.

Password : The password needed to check the validity of the new date file.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : 1 if the function is successfully completed.

See also: Date stamping .

DaysGoneBy function

```
function DaysGoneBy (FileName, password :PChar; v :byte) :integer; far; external
'SWTOOLS' index 4;
```

Description:

Determines the number of days that have passed since the program was first run, by checking the date stamp contained in *FileName* and comparing it to the current date. If the date stamp file is empty , it will create a new one with the current date.

Parameters:

FileName : The name and path of the date stamp file.

Password :The password to be used to check the validity of the date stamp and, if the file is /popoplink EMPTY,empty , to create the new one.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : The number of days since the stamp in the specified file. If it cannot find the file, it will return 999; if the password is invalid, it will return 888.

See also: Date stamping .

CreateDateFile function

```
function CheckRegistration (FileName, Name, Password :PChar; more :integer; var
MoreToFollow :PDosStream; v :byte) :integer; far; external 'SWTOOLS' index 5;
```

Description:

Checks a registration file and determines if it is valid or not. It also prepares the file for GetRegInfo.

Parameters:

FileName : The path and name of the registration file.

Name : This name of the registered user will be placed wherever this pointer points.

Password : The password which was used to crete the registration file. It is used to make sure the registration file was really created by your program.

More : Set this equal to 0 if no more information is to be read in from the file; in that case, it closes the file itself. Otherwise, the file is left open for GetRegInfo, and must be closed with DoneWithInfo.

MoreToFollow : A pointer to a DOS stream, which is passed to GetRegInfo to get more information and to DoneWithInfo when you are done with the file.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : 1 if the registration file is there and is valid; 0 if it isn't.

See also: Registration tools .

CreateRegistration function

```
function CreateRegistration (FileName, Name, Password :PChar; more :integer; var
MoreToFollow :PDosStream;   v:byte) :integer; far; external 'SWTOOLS' index 6;
```

Description:

Creates a new registration file by scrambling *Name* and *Password* and storing it in *FileName* . Also prepares the file for more information to be written to it by WriteRegInfo.

Parameters:

FileName : The path and file name of the registration file to be created. Note that CreateRegistration does not add any extension to the file name.

Name : The name of the person registering; it is scrambled with the password and stored in the file. Any characters beyond the tenth will be ignored.

Password : The password used to scramble name. Later, the same password should be passed to CheckRegistration.

More : Set this equal to zero if you want to write just the name and registration number (name scrambled with password) to the file; in that case, it will be closed automatically. Otherwise, it is left open so that more data can be written by WriteRegInfo. If you leave it open, be sure to close it with DoneWithInfo when you are done.

MoreToFollow : Will return a pointer to a DOS stream to be passed to WriteRegInfo and DoneWithInfo if *More* is not zero.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : The number passed in *More* .

See also:

Registration tools .

GetRegInfo function

```
  function GetRegInfo (MoreFollowing :PDosStream; var ExtraInfo :PChar;
v :byte) :integer; far; external 'SWTOOLS' index 7;
```

Description:

Retrieves an additional piece of information from the file pointed at by MoreFollowing. This pointer should be created by CheckRegistration.

Parameters:

MoreFollowing : A pointer to a DOS stream; You should always pass the return from CheckRegistration.

ExtraInfo : The string retrieved will be placed wherever this pointer points.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : The length of ExtraInfo.

See also:

Registration tools .

WriteRegInfo function

```
function WriteRegInfo (MoreFollowing :PDosStream; ExtraInfo :PChar; v :byte):integer;
far; external 'SWTOOLS' index 8;
```

Description:

Sends an additional piece of information to the file pointed at by MoreFollowing. This pointer should be created by CreateRegistration. Note that the pieces of information must be no more than ten characters long.

Parameters:

MoreFollowing : A pointer to a DOS stream; You should always pass the return from CheckRegistration.

ExtraInfo : Should point to a string, which will be written to the registration file.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : The length of ExtraInfo.

See also:

Registration tools .

GetRegInfo function

```
function Scramble (MoreFollowing :PDosStream; v:byte):integer; far; external
'SWTOOLS' index 9;
```

Description:

Properly closes the file pointed at by *MoreFollowing* , and frees all associated memory. Should alway be called after calling CheckRegistration or CreateRegistration with the *More* parameter nonzero.

Parameters:

MoreFollowing : A pointer to a DOS stream; You should always pass the return from CheckRegistration or CreateRegistration.

v : The version number of SWTOOLS you are using. This parameter is provided so that programs written using earlier verions will still be compatible with the updates.

Returns : 1 if the function is completed.

See also: Registration tools .

Date stamping

Date stamping is what I call the process SWTools uses to keep track of how long you program has been used for. It involves creating a file which contains the date your program was first executed, along with a code to verify that your program put the date there. Then, the next time the program is run, it reads in this file and determines how many day have passed.

To use date stamping, all you have to do is call the DaysGoneBy function . If this function finds an empty date file , it will write over it with the current date, and report back that 0 days have passed. After that, it will return the number of days that have elapsed since it created the date file. It is important to distribute your program along with an empty date file; otherwise, DaysGoneBy will return an error when it is called.

Because SWTools keeps track of the number of days that have passed since an empty date file was read, and you have to include an empty date file with your program package, it is possible to work around the date stamping technique by simply unZIPping the file again every time the time limit has passed. However, having to go through the process of doing this every 30 days should be enough to at least make them think about registering. Besides, many people delete their ZIP files, or wouldn't think to unZIP them again. And the reason many people don't register is they don't ever really think about it; having to unZIP the file again will force them to.

See also: CreateDateFile function , CreateNewDate function , DaysGoneBy function

Registration tools

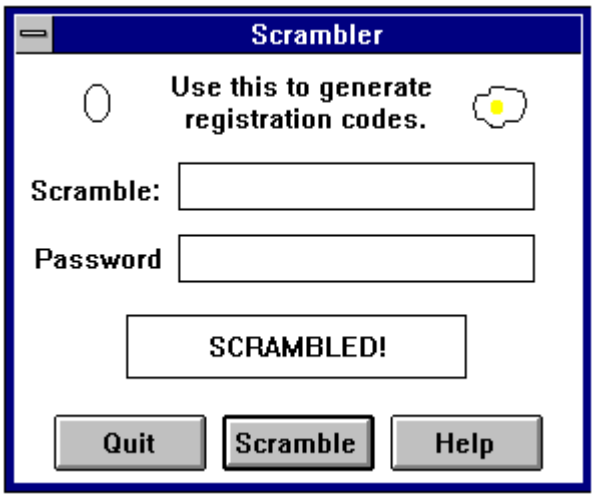
The registration tools create and handle a registration file, which contains the name of the person registered and a registration code. It can also contain extra pieces of information, such as default settings for your program.

These tools simplify the registration process a great deal. Instead of you having to send every registered user a copy of the registered version of your program, all you have to send them is a registration code. Registration codes can be generated with the Scrambler tool .

To create a registration file, use the CreateRegistration function . To check a registration file's validity, use CheckRegistration .

See also: CheckRegistration function , CreateRegistration function , GetRegInfo function , WriteRegInfo function , DoneWithInfo function .

Scrambler tool (SCRAMBLE.EXE)

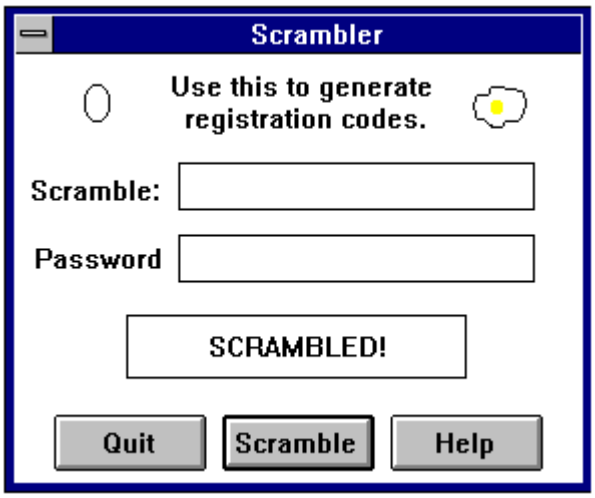


When it is first started, Scrambler looks like above. To use it, type the word to be scrambled in the box labeled Scramble: and the password to scramble it with in the Password box. Press the Scramble button, and the result will appear in the box which now says SCRAMBLED! .

To use this to generate registration codes, type in the person's name in the Scramble: box and you program's password in the Password box. The result will be the registration code.

All this tool does is call the Scramble function , with the contents of the Scramble: and Password boxes as the parameters.

MakeDate tool (MAKEDATE.EXE)



When it is first started, MakeDate looks like the above. It is used to generate empty date files , which will change to date stamps when read by DaysGoneBy . To use it, type in the name, path, and extension of the empty file you wish to create in the File box, and your program's password in the Password box. When you push the Create button, the file will be created.

This tool simply calls CreateNewDate with the contents of the boxes labeled Filename and Password as the parameters.

Copyright and license information

License

If you have not registered this program, you are licensed to use it for an unlimited amount of time on your own computer AS LONG AS YOU DO NOT DISTRIBUTE PROGRAMS WHICH USE SWTOOLS.DLL. If you wish to use this program as a part of a distributed software package (and it really isn't very useful if you only use it yourself), you must get it registered.

If you are a registered user, you are liscensed to use all of the functions contained in SWTOOLS.DLL in your own programs, and to distribute SWTOOLS.DLL along with your software package. You are also licensed to use parts of the demo program in your own programs. And thank you for registering.

All users, both registered and unregistered, are licensed, and encouraged, to distribute this program, by passing it along to friends or uploading it to BBS's, as long as they include with it ALL of the files it originally came with. It is recommended you simply pass on the original ZIP file.

Disclaimer

To the best of my knowledge, ShareWare Tools V1.0 functions exactly as it is described in this help file. However, it is distributed as is, with no warranty of any kind.

See also: Registration .

Registration info

The cost to register this progam is $10 per copy. Only one copy need be registered per person or organization, but you cannot pass your registration privelages on to another person. If this program is to be used by more than one programmer or software company, each one must have their own registration.

Registration entitles you to notice of major updates (basically, any update that affects the program more than cosmetically), and to recieve copies of the updated versions for the cost of shippig and handling. For $20, you will be automatically sent all updated versions at no additional cost. All registered users get unlimited customer service, either by electronic mail or by regular mail.

If you've gone to the trouble of aquiring this program, you are probably a ShareWare programmer. If you are a ShareWare programmer, you probably understand how important it is to register programs you use. For this reason, I have decided not to include the normal groveling paragraph, and just ask you nicely to register. So here goes.

Please register this program.

To register: Registration form

ShareWare Tools V1.0 Registration

To print out a copy of this registration form, go to File/Print Topic in the menu above. Please feel free to write your suggestions, comments, or compliments (compliments are especially appreciated!) on the bottom of this form.

Name:_____

Address:_____

City, state:_____

ZIP code:_____

E-mail address:_____
(specify Internet, CompuServe, AOL, or Prodigy;
If you don't have one, leave this blank.)

_____ Basic registration ($10)
(includes notice of all updates, unlimited customer assistance, my undying gratitude, and a really nice Thank-You note.)

_____ Extended registration ($20)
(includes free copies of all updates, unlimited customer assistance, my undying gratitude, and TWO really nice Thank-You notes.)

Please make checks payable to Scott Gifford.

Pease send this form, along with check or money order for the amount specified above, to:

Scott Gifford
1014 Drury Lane
Flushing, MI 48433

Thank you!

About the Author

You can contact me, Scott Gifford, electronically at the following addresses:

CompuServe: 71234,1147
Prodigy: JXPV11F
Internet: 71234.1147@compuserve.com
ottscay@umcc.ais.org
America OnLine: ottscay

or by paper at:

Scott Gifford
1014 Drury Lane
Flushing, MI 48433

Empty date files are files created by the CreateNewDate function or by the MakeDate tool . These are the files you should distribute with your program. When DaysGoneBy reads this file, it writes over it with the current date.

Passwords are used by many of the functions in SWTools to validate information, such as the date stamp or the registration information. The password should be up to eight characters long and defined in your program. If you change the password you use, SWTools will find all files created with the old one invalid.

Note to non-TPW programmers

This DLL was compiled using Turbo Pascal for Windows V1.5. Because of this, everything in this help file, and in the demo, refers to Pascal. If you are programming in another language, you may have to make some changes, although they should be minor.

One important thing to note is two types used in this DLL:PChar and PDosStream. The first refers to a pointer to a null-terminated string, and the second to a pointer to a DOS stream. If the language you are using does not define types like these, just use plain old pointers.

If you have problems using this DLL from another language, please contact me at one of the addresses listed under Problems, Questions, and Comments and I will do my best to help.